Git/GitHub/Eclipse

Lesson 8

Vicki Long and Tenzin Ukyab

Git?

Its a Version Control System (VCS)

- keep track of changes you make to your files
- go back to older versions of your code

On Windows, git uses BASH (Bourne Again SHell) which is a command-line interpreter.

On Macs you use the Terminal.

Command line?

A CLI (command line interface) allows you to type in text commands to interact with the computer.

To use Git, open Git Bash. A terminal will open up, in which you can type in commands.

Fundamental commands

ls

- lists all the files in the folder you are in
- make sure you are in the right folder

cd [insert name of folder or directory here]

- moves you into the folder you want to work in

More commands

cd ..

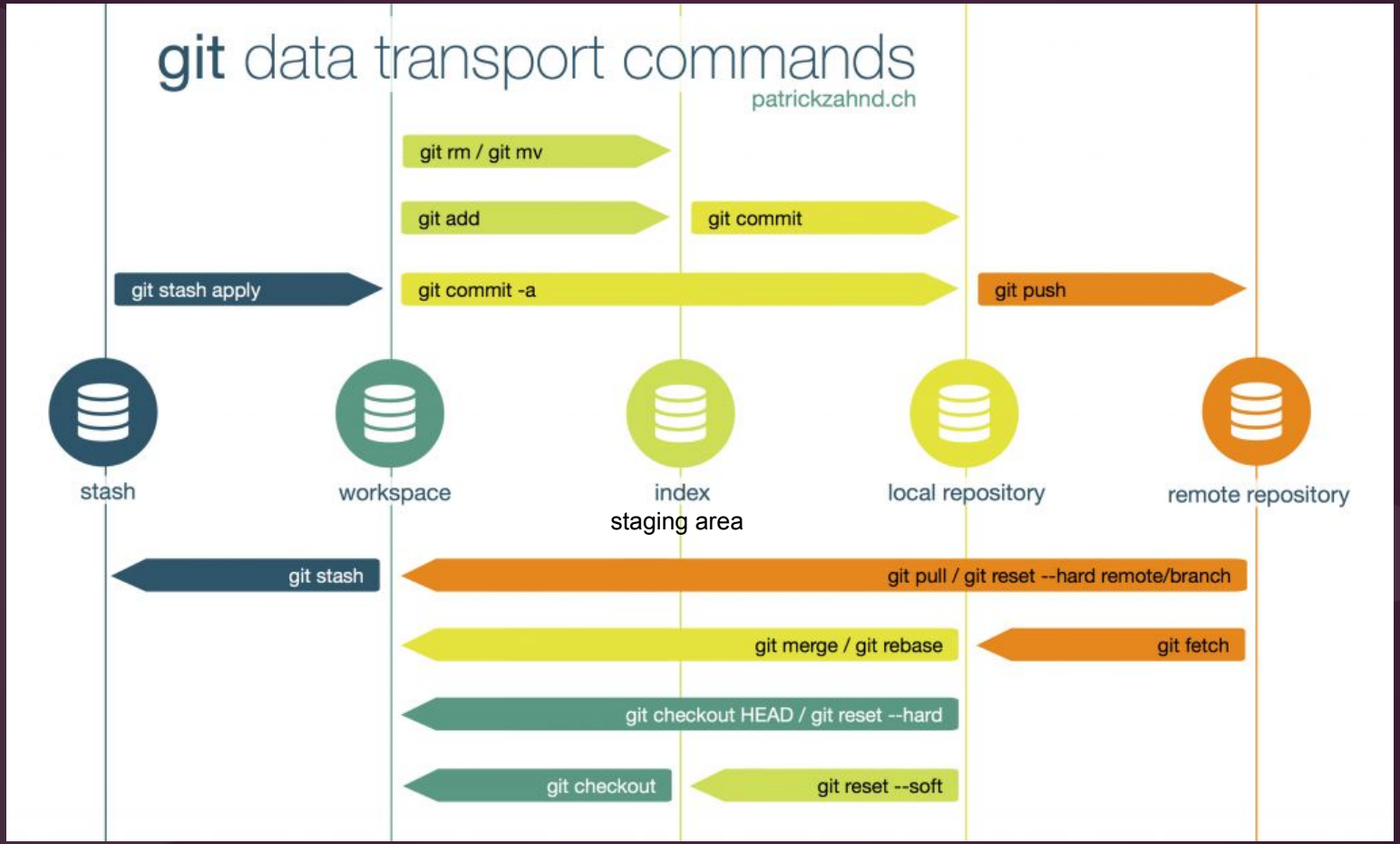
- go out one folder

cd ../..

- go out two folders

GitHub?

Allows lots of people to collaborate on code together.


git data transport commands
patrickzahnd.ch
git rm / git mv
git add
git commit
git stash apply
git commit -a
git push
stash
workspace
index
staging area
local repository
remote repository
git stash
git pull / git reset --hard remote/branch
git merge / git rebase
git fetch
git checkout HEAD / git reset --hard
git checkout
git reset --soft

The folder on your computer.

The staging area.

The HEAD.

GitHub

Signing in

1. git config --global user.name “Your username”
2. git config --global user.email your@email.com

New repo from compu

git init

- allows Git to keep track of all files and file changes in the current folder
- cd to the folder before hand

Go to GitHub, make a new repository and follow the instructions.

Clone repo from GitHub

git clone [paste URL here]

- creates a new folder with all the files on a repo already on GitHub
- make sure you are currently in the folder you want the files to be in
- no need to initialize

Checking stuff

git status

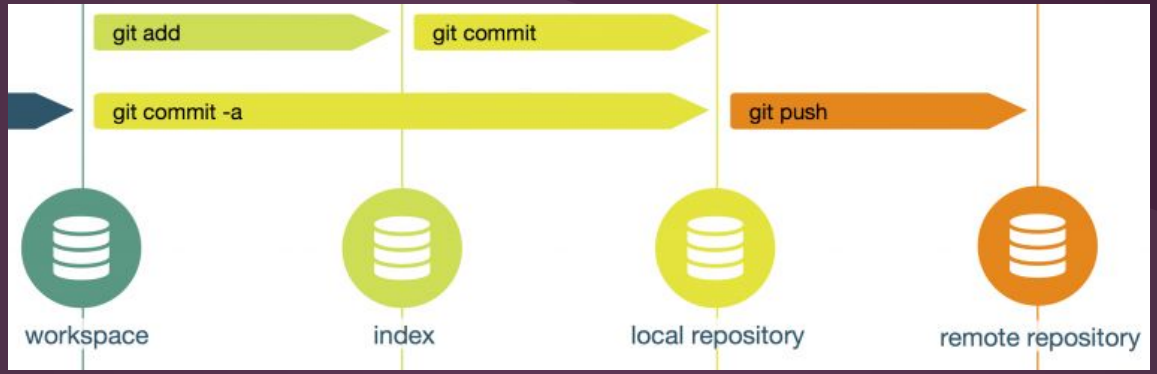
- checks if your workspace is up to date with the remote repo (RECOMMENDED)

git diff

- shows the difference

git log (--help for a more refined search)

- shows all previous commits, messages, etc


git add
git commit
git commit -a
git push
workspace
index
local repository
remote repository

Adding to the index

git add [file name]

- Adds the file to staging area
- (RECOMMENDED)

git add .

- Adds all untracked files to staging area

Committing to your local repo

git commit [file name]

- opens a prompt to type in a message

git commit -a

- stages all changes and open prompt for message

Committing (cont)

git commit -m “[message here]” [file name]

- type in a message for a specific file without the prompt

git commit -am “[message here]”

- stage and type a message for all edited files without the prompt (RECOMMENDED)

Commit messages

Commit messages are an important way to track changes.

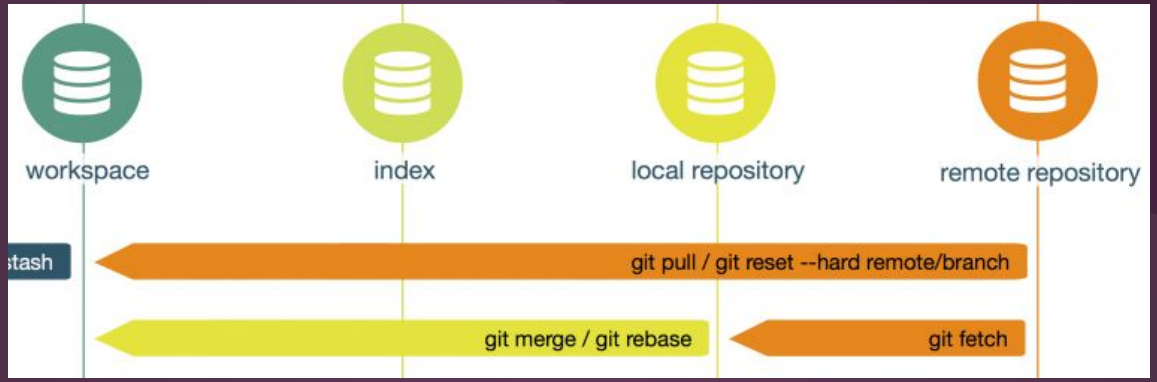
- start with an active verb
- present tense
- concise
- specific

Pushing to the remote repo

git push [alias] [name of your local branch]

- pushes your changes to GitHub
- alias is usually origin
- also can do just “git push”

Prompts you to enter your username and password.


workspace
index
local repository
remote repository
tash
git pull / git reset --hard remote/branch
git merge / git rebase
git fetch

Pulling

git pull

- Pulls changes from GitHub if your files aren’t up to date
- Should be done before pushing your changes

Conflicts

After pulling changes from GitHub or merging two branches, a conflict may happen.

This occurs because you made a change to specific line that was changed in another commit.

Conflicts (cont)

Open the file where the conflict is.

You will see…

<<<<<< where the conflict begins

>>>>>> where the conflict ends

======= between the two conflicts

Conflicts (cont)

- Delete the lines you don’t want
- Make proper corrections
- Save
- Stage
- Commit
- Push

Conflict example

Here, the change I made was line 44. What I pulled from GitHub was line 46.

```
39              <ul class="nav navbar-nav">
40                <li class="dropdown">
41                  <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button"
                     aria-expanded="false">FIRST<span class="caret"></span></a>
42                  <ul class="dropdown-menu" role="menu">
43  <<<<<<< HEAD
44                    <li><a href="aboutFirst.html">About</a></li>
45  =======
46                    <li><a href="AboutFirst.html">About</a></li>
47  >>>>>>> 1d1f9bc39e77a69a0bca157ea0bd180da84df463
48                    <li><a href="#">2015 Game</a></li>
49                    <li><a href="#">Past Games</a></li>
50                  </ul>
51                </li>
```

To resolve the conflict, I would delete lines 43 and 45-47. Then I would save, commit, and push.

Resetting

git reset [file]

- Unstages the file, but preserve its contents

git reset [commit]

- Undoes all commits after [commit], preserving changes locally

Resetting (cont)

git reset --hard [commit] / remote / [branch]

- Discards all changes and resets workspace back to that certain time

For [commit] you only have to write the first 5 or 6 characters of the commit. If the commit is aba0b7ddb…, use “git checkout aba0b7”.

Branches

A branch in Git is generally used when you want to work on something new and not add to the main code.

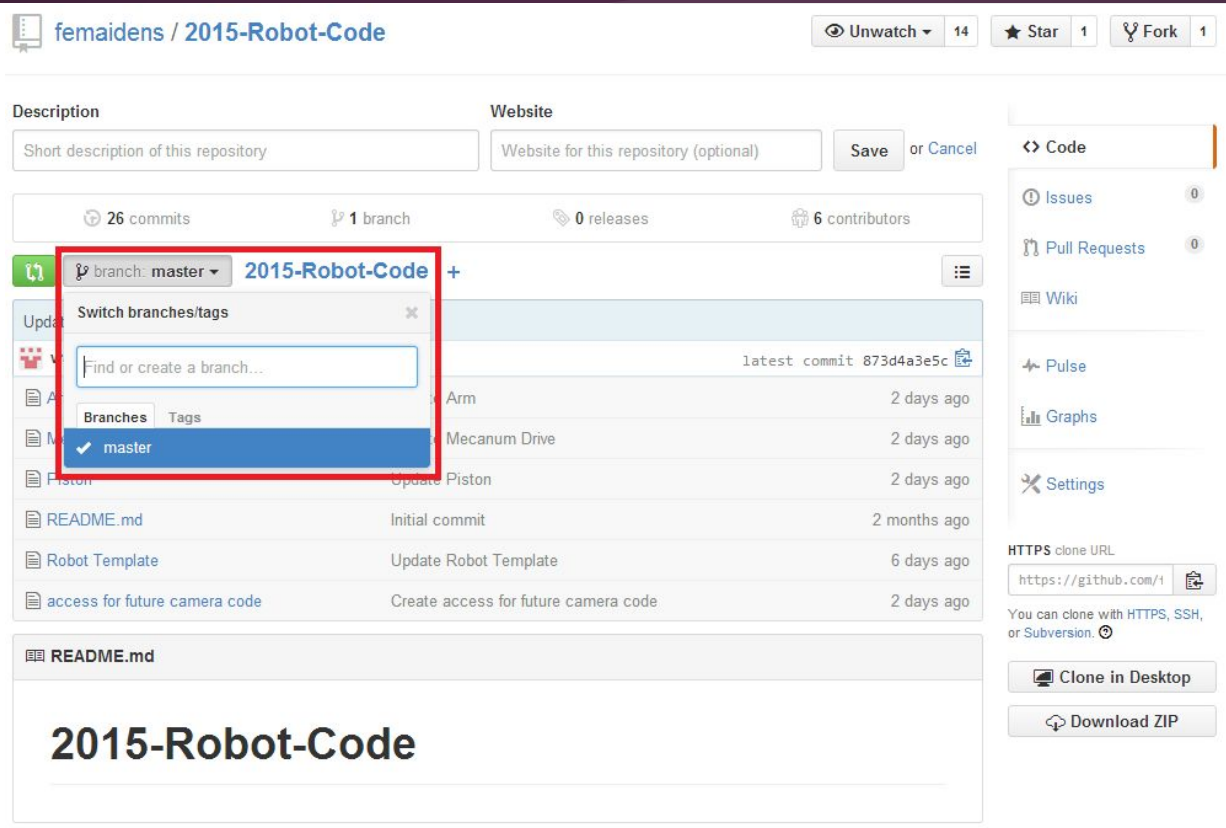
For example, if you have robot code that works, you don’t want to mess it up with untested code, so you would make a new branch to write untested code in.

Branches (cont)

Master branch

- When you first initialize a repo, this is the branch it has
- Usually considered the main branch
- Should contain most recently tested and working stable code.

A branch on GitHub



Branch commands

git branch [name of branch]

- make a new branch

git branch

- lists all your branches

git branch -d [name of branch to delete]

- removes the branch
- recommended after merging

Merging

git merge [name of the branch to merge with]

- joins the two branches together
- make sure you are in the branch you want the merge to be applied to
- this is best used when you want to merge an experimental branch with your main branch

Rebasing

git merge

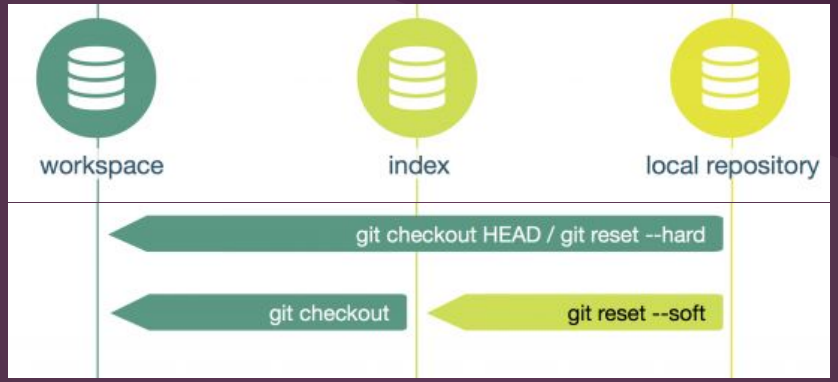
- make sure you are in the branch you want the rebase to be applied to
- this is best used when you want to apply changes from your main branch onto an experimental branch

Fetching

git fetch origin

- moves all commits from the remote repository to your local repository
- fetch before you merge or rebase

Don’t have to use this much.


workspace
index
local repository
git checkout HEAD / git reset --hard
git checkout
git reset --soft

Switching branches

git checkout [name of branch]

- Switches to another branch

git checkout -b [name of branch]

- make a new branch and switch to it at the same time

Going back to old code

git checkout [commit]

- changes the state of your repo to what it was when you made that commit

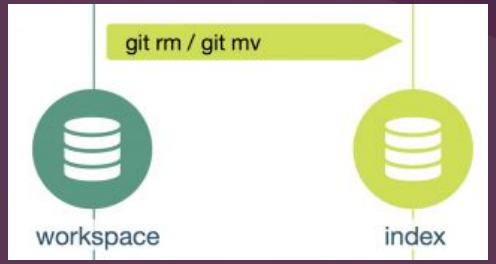
git checkout [name of branch you want to go back to]

- switch back to the current version of your code

Returning to current code

git checkout [name of main branch]

- switch back to the current version of your code
- generally will be “git checkout master” because master branch is usually the main branch


git rm / git mv
workspace
index

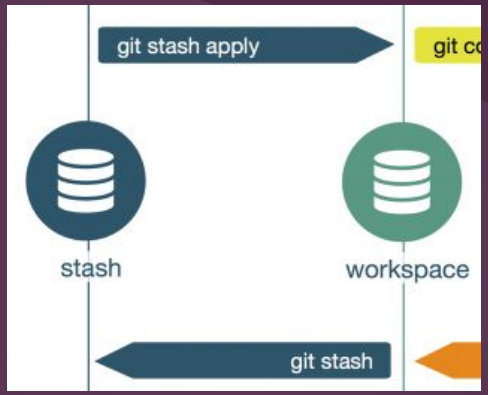
Refactoring files

git rm [file name]

- Deletes the file from the working directory and stages the deletion

git mv [file-original] [file-renamed]

- Changes the file name and prepares it for commit


git stash apply
git co
stash
workspace
git stash

Stashing

If you want to save changes you made to one branch but not commit them and work on another branch, you can stash them so that you can get to them later.

git stash

- Temporarily stores all modified tracked files

Stashing (cont)

git stash list

- Lists all stashed changesets

git stash pop (or apply)

- Restores the most recently stashed files

git stash drop

- Discards the most recently stash

Review: Pushing

git pull

Check for conflicts and fix them.

git status

git add [file-name] x how many files you have edited

git status

git commit -am “good commit message here”

git status

git push

Check on GitHub if actually pushed.

Review:

Pulling: git pull

Switching branches: git checkout [branch-name]

If you want to do anything else refer back to this ppt or ask me/Google for help.

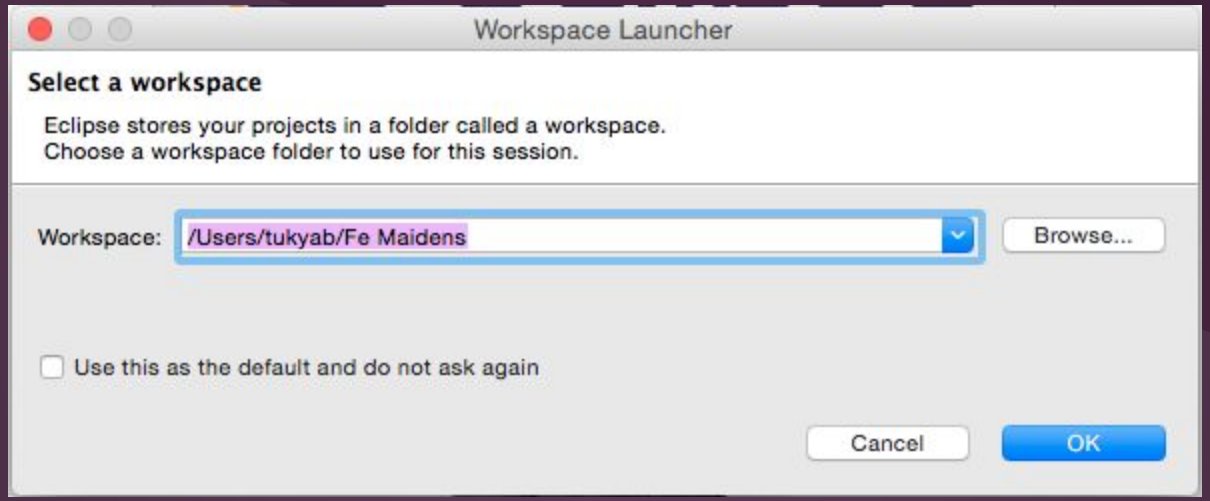
Eclipse?

Its an Integrated Development Environment (IDE) or an editor.

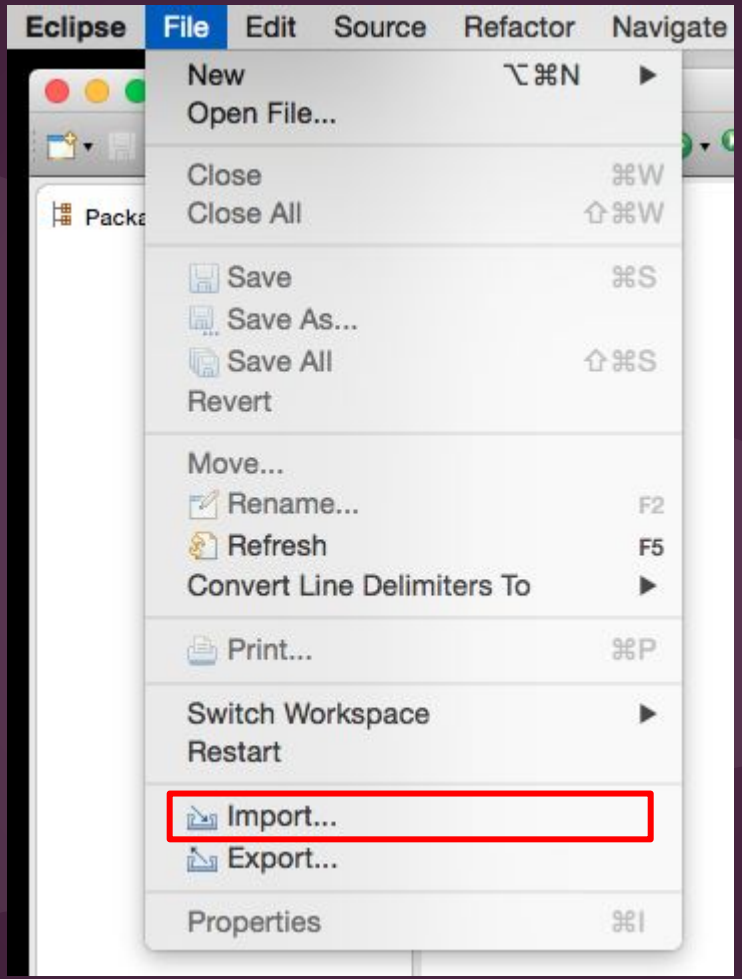
We will be working with Eclipse to edit our code.

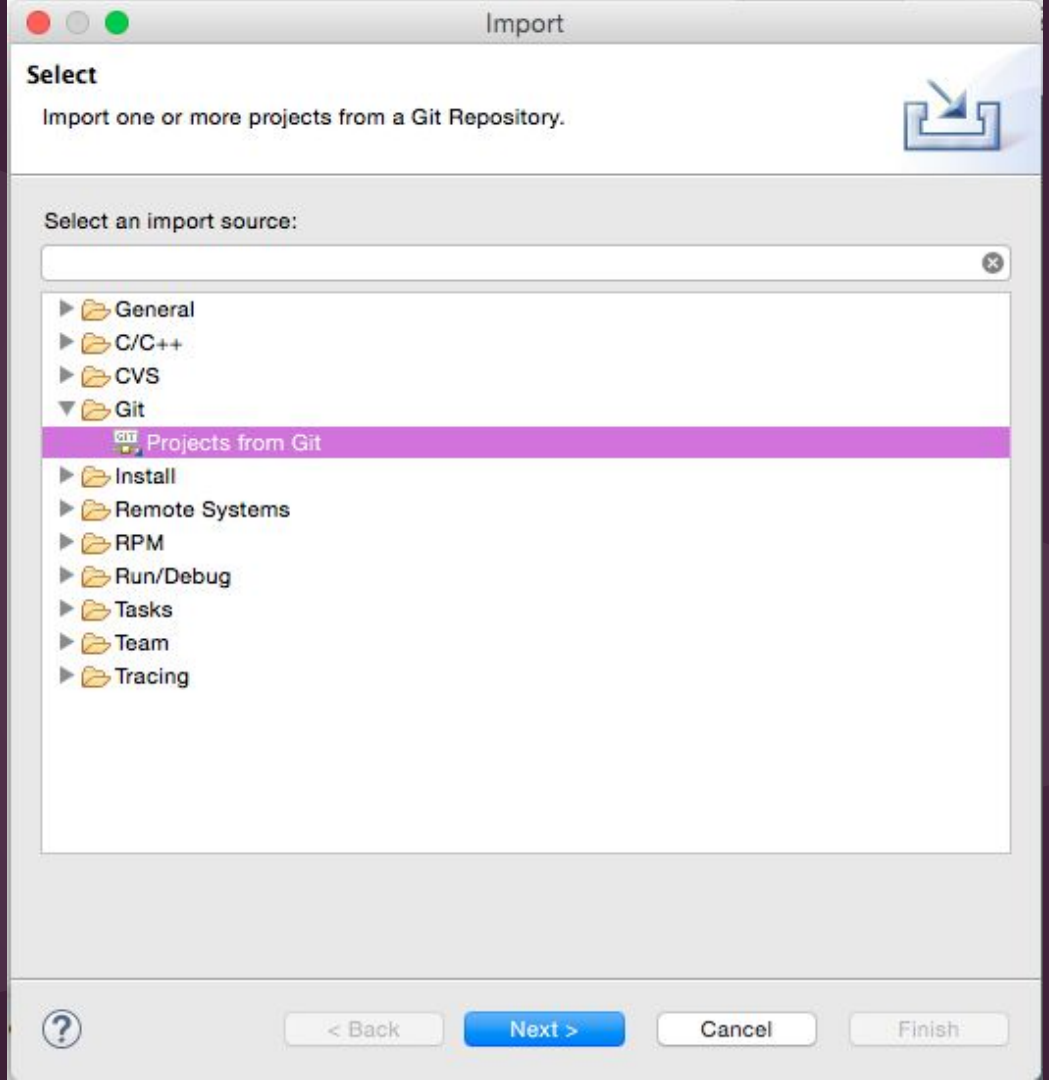
The next slides show how to get your project into Eclipse.

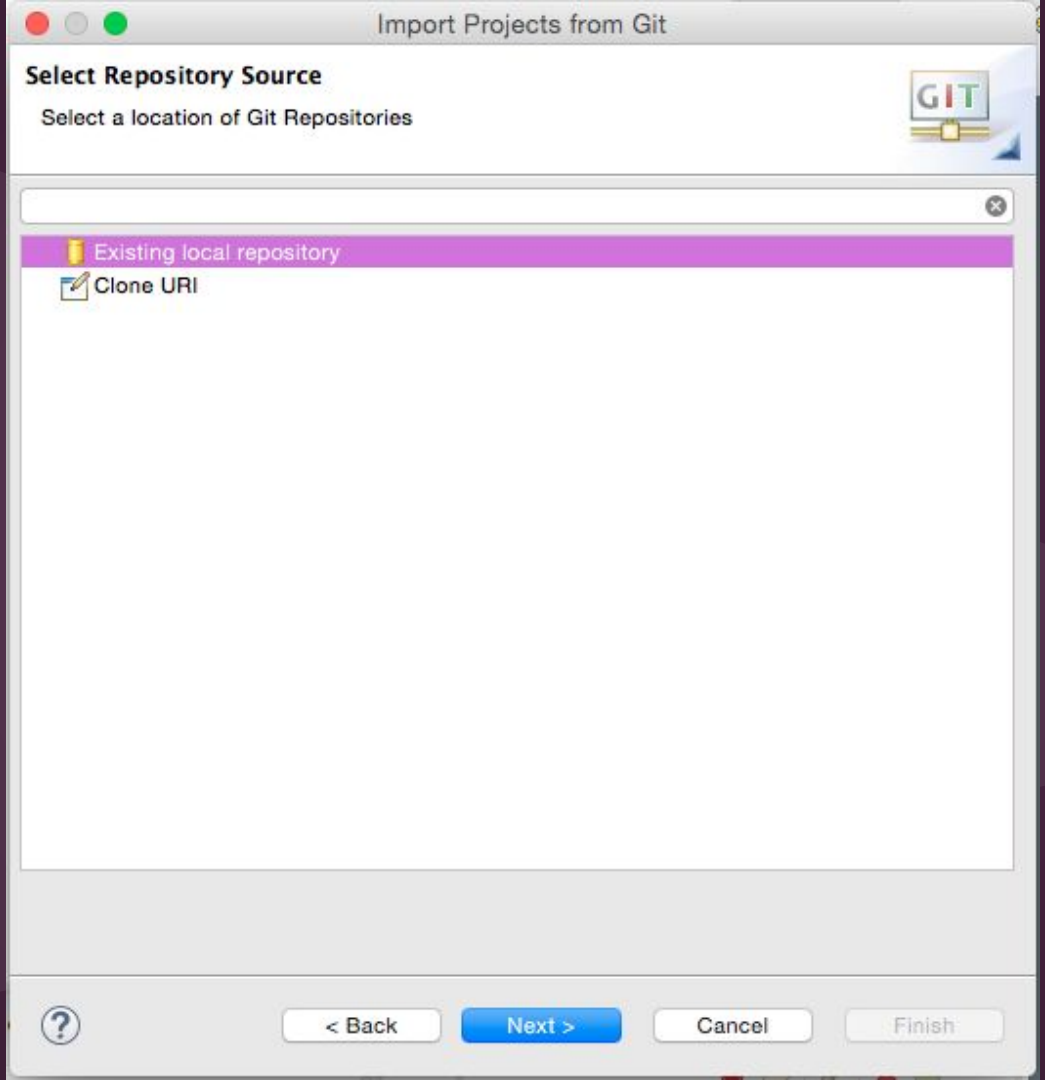
Open up Eclipse

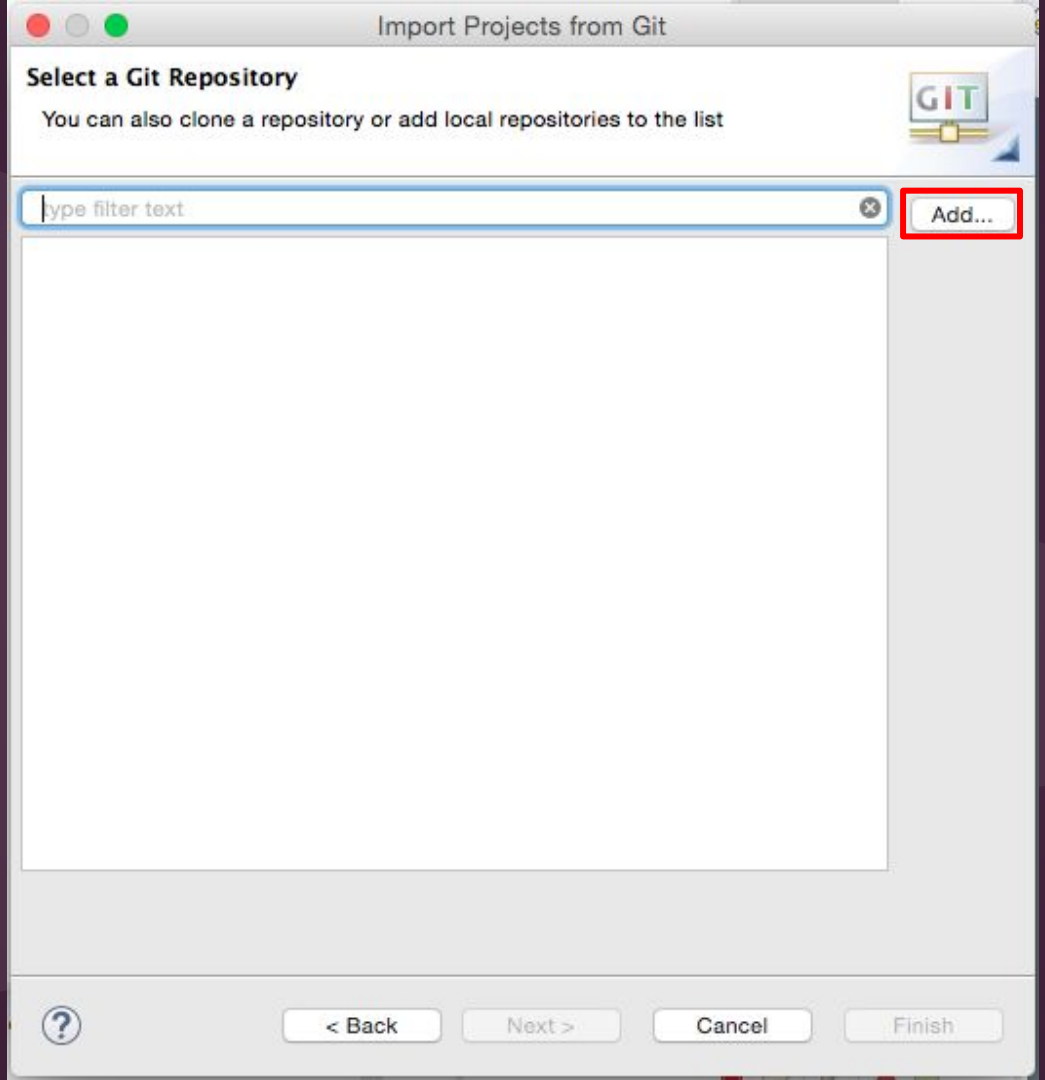


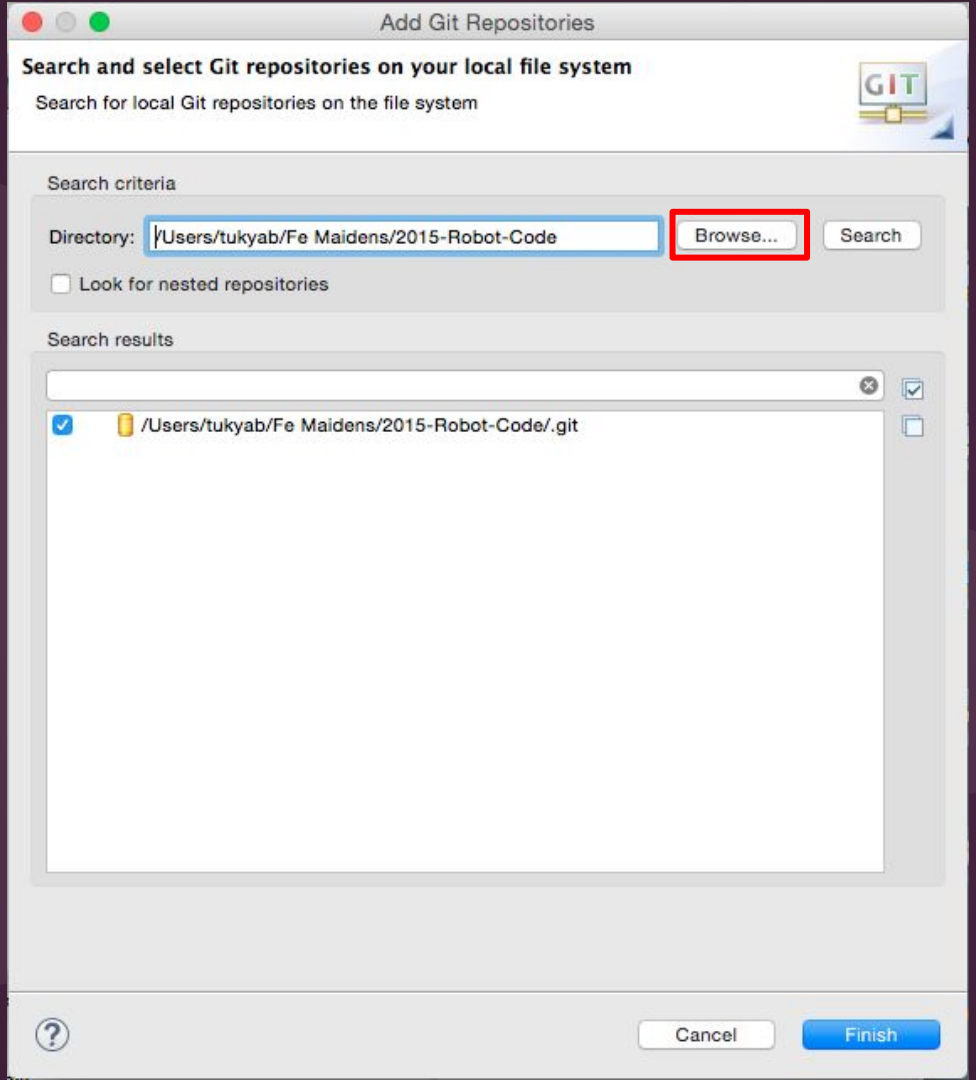
Click Browse to select your workspace.

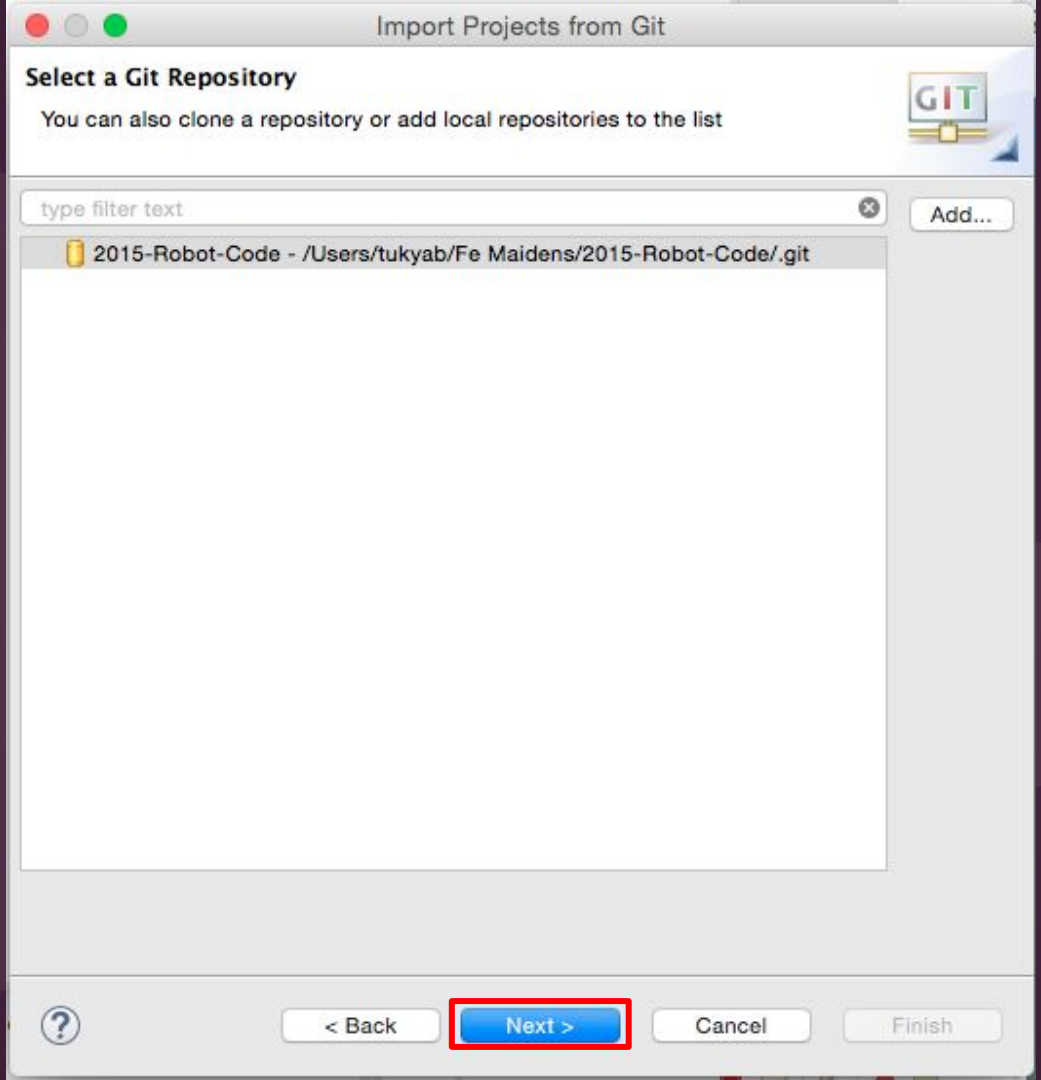

Eclipse
File
Edit
Source
Refactor
Navigate
New ⌥⌘N
Open File...
Close ⌘W
Close All ⇧⌘W
Save ⌘S
Save As...
Save All ⇧⌘S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... ⌘P
Switch Workspace
Restart
Import...
Export...
Properties ⌘I

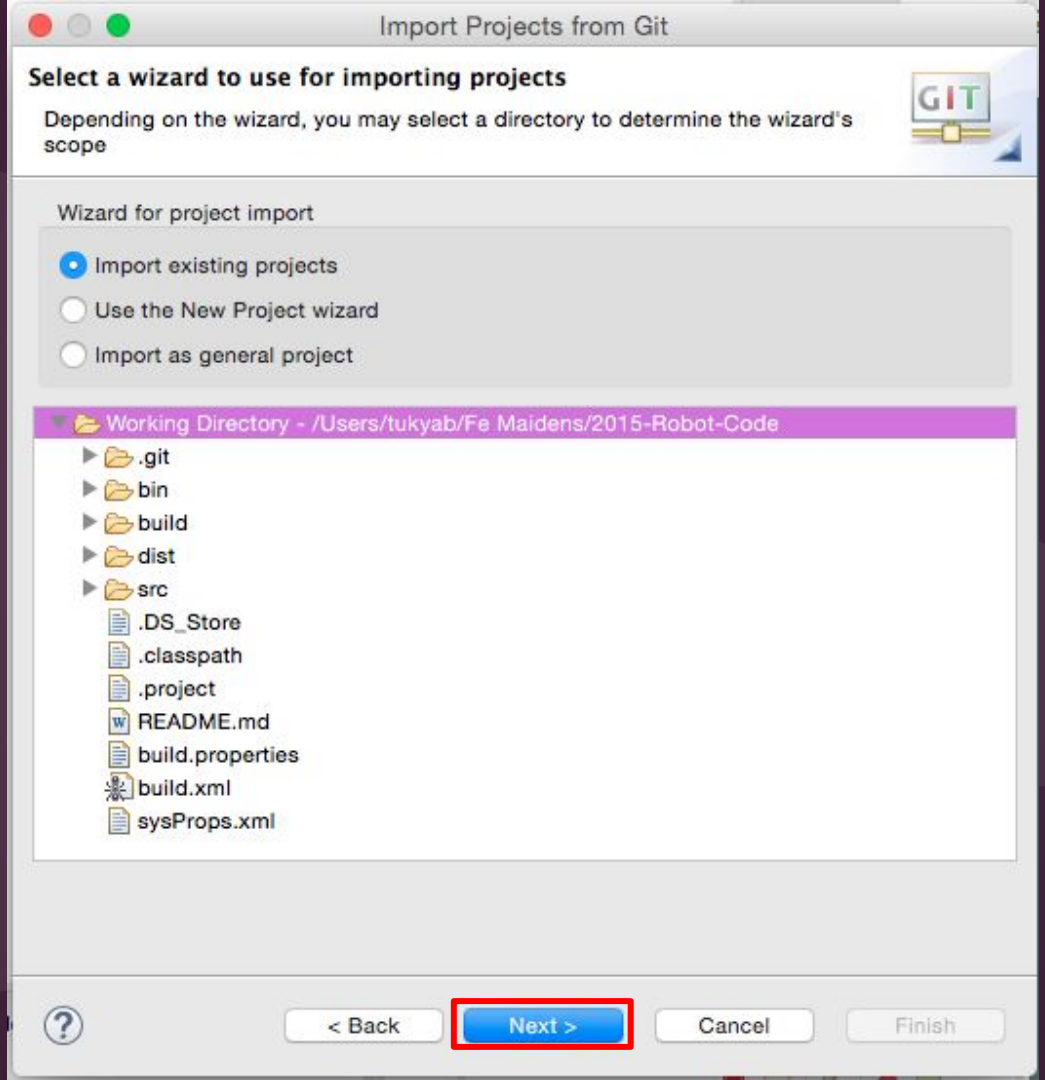

Import
Select
Import one or more projects from a Git Repository.
Select an import source:
General
C/C++
CVS
Git
Projects from Git
Install
Remote Systems
RPM
Run/Debug
Tasks
Team
Tracing
< Back
Next >
Cancel
Finish

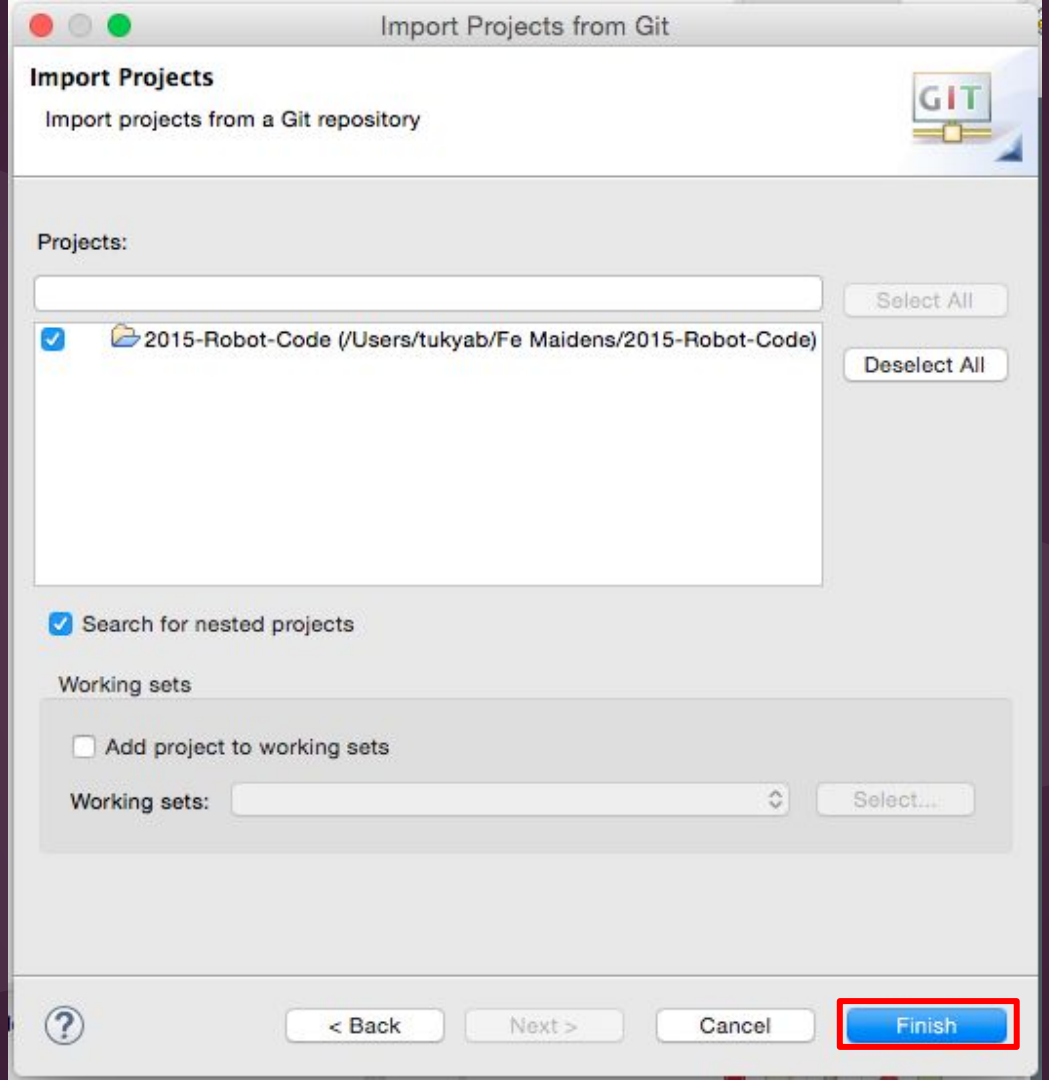

Import Projects from Git
Select Repository Source
Select a location of Git Repositories
GIT
Existing local repository
Clone URI
< Back
Next >
Cancel
Finish


Import Projects from Git
Select a Git Repository
You can also clone a repository or add local repositories to the list
GIT
type filter text
Add...
< Back
Next >
Cancel
Finish

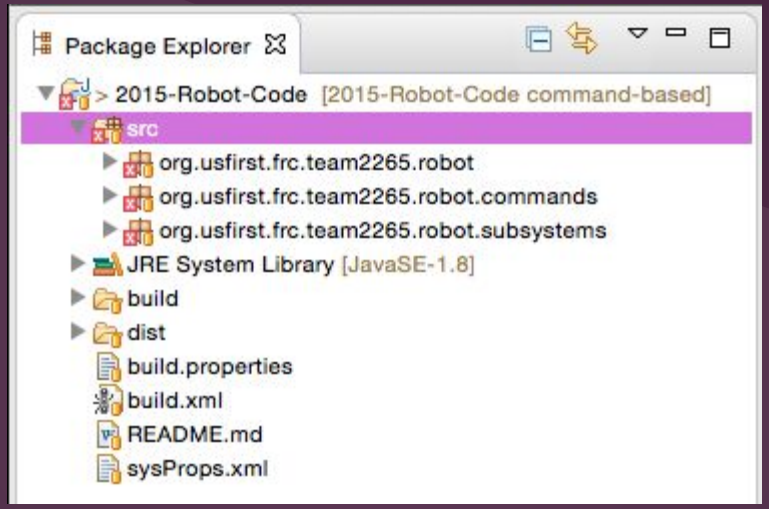

Add Git Repositories
Search and select Git repositories on your local file system
Search for local Git repositories on the file system
GIT
Search criteria
Directory:
/Users/tukyab/Fe Maidens/2015-Robot-Code
Browse...
Search
Look for nested repositories
Search results
/Users/tukyab/Fe Maidens/2015-Robot-Code/.git
Cancel
Finish


Import Projects from Git
Select a Git Repository
You can also clone a repository or add local repositories to the list
GIT
type filter text
Add...
2015-Robot-Code - /Users/tukyab/Fe Maidens/2015-Robot-Code/.git
< Back
Next >
Cancel
Finish


Import Projects from Git
Select a wizard to use for importing projects
Depending on the wizard, you may select a directory to determine the wizard's scope
GIT
Wizard for project import
Import existing projects
Use the New Project wizard
Import as general project
Working Directory - /Users/tukyab/Fe Maidens/2015-Robot-Code
.git
bin
build
dist
src
.DS_Store
.classpath
.project
README.md
build.properties
build.xml
sysProps.xml
< Back
Next >
Cancel
Finish


Import Projects from Git
Import Projects
Import projects from a Git repository
GIT
Projects:
Select All
2015-Robot-Code (/Users/tukyab/Fe Maidens/2015-Robot-Code)
Deselect All
Search for nested projects
Working sets
Add project to working sets
Working sets:
Select...
< Back
Next >
Cancel
Finish

Result:


Package Explorer
2015-Robot-Code [2015-Robot-Code command-based]
src
org.usfirst.frc.team2265.robot
org.usfirst.frc.team2265.robot.commands
org.usfirst.frc.team2265.robot.subsystems
JRE System Library [JavaSE-1.8]
build
dist
build.properties
build.xml
README.md
sysProps.xml

HW: Download Git + Eclipse

https://git-scm.com/downloads

Windows: Git BASH

Mac: Terminal

Eclipse

Learn MORE Git

- Tutorials: The Simple Guide (basic), Git Immersion (full length), Codeacademy (lesson based), YouTube Videos, Guides (in depth), Confused?
- Documentation: Reference, Pro Git Book (in depth), Glossary, Visual Git Reference
- Try Git Online: Try Git (short), Git Real (longer)
- Git Cheat Sheets in this folder